Research Track 2 - First Assignment

Enrico Piacenti - S5636699

Robotics Engineering, at University of Genova, Genova, Italy.

Address correspondence to: enricopiacenti96@gmail.com

01/06/2023

Abstract

The objective of this assignment is to conduct a comparative analysis between two algorithms: one developed by myself and the other by my colleague. These algorithms are designed to accomplish a specific task. Additionally, a statistical comparison is performed based on an initial hypothesis proposed by me.

To compare the two scripts, various factors are taken into consideration, such as the way tokens are placed in the arena, the average execution time, and the proportion of failures to successes. These metrics help determine the performance of each algorithm under different conditions. In order to obtain statistically significant results, a t-test is conducted by evaluating the values obtained from multiple executions of each algorithm.

Overall, the purpose of this assignment is to evaluate the relative performance of the two algorithms and identify the conditions under which each algorithm excels, utilizing statistical analysis to ensure the validity of the findings.

1 Introduction

1.1 Environment

Each of our scripts were run in two different scenarios in order to compare how well my algorithm and my colleague's performed.

- Placing tokens on the two circles: Evaluate how effectively the algorithms work when tokens are placed on the two circles.
- Randomly distributed tokens in the environment: Evaluate how well the algorithms function when tokens are dispersed randomly across the environment.

The number of test were 50 whereas the number of tokens was three for each color (so three silver and three gold).

1.2 Data measurements

A short bash script was written to retrieve all the necessary data and automate all the tests by running the same script 50 times. The expiring time was arbitrarily set at 300 seconds, killing all threads created after this selected time. the two measurements that were made are:

- The success/failure amount Keep track of how many times each algorithm has been used to recover tokens successfully and unsuccessfully. An attempt is deemed successful if every token is retrieved, but an attempt is deemed unsuccessful if one or more tokens are still missing. This metric will be used to assess the algorithms' efficiency and robustness.
- Estimated Time Needed Calculate how long it takes on average for each implementation to finish the task of obtaining all tokens. This statistic will provide insight on the algorithms' effectiveness in terms of time complexity.

1.3 Hypothesis Formulation

Starting from the premise that there is no difference between the two algorithms, I have formulated a null hypothesis (H0) , additionally another alternative hypothesis is set based on the premise that there are some differences in performance. These are my presumptions as a result:

- Null Hypothesis (H0): There is no difference between algorithms in terms of time execution.
- Alternative Hypothesis (H1): There are differences between algorithms in terms of time execution.

1.4 T-test to analyze the Results

In order to do a statistical analysis to get to a conclusion, a t-test is made. The t test determines how significant the differences in group means are. It reveals whether those disparities in means could have arisen by accident. When data sets have a normal distribution but the population variance is unknown, the t test is typically used. In other words, by applying the t-test to the values returned by my algorithm (time complexity) and that of my colleague, it is possible to determine whether the algorithms' performances are comparable or whether one algorithm outperforms the other. It is important to note, however, that these values must take into account the success percentage of each script in the 50 tests, of course not only time matters.

2 Results

When the bash script is launched, the python script is automatically executed 50 times, and the mobile robot data is obtained. All of the values obtained are listed in the table below, along with whether or not the Task was successful:

[**Table of Values obtained from the tests**]

	Token Randomly Generated		Token Circularly Generated	
	Execution Time	Success	Execution Time	Success
Test 1	80	Yes	126	Yes
Test 2	93	Yes	79	Yes
Test 3	195	No	79	Yes
Test 4	172	Yes	108	Yes
Test 5	118	Yes	131	Yes
Test 6	113	Yes	89	Yes
Test 7	78	No	128	Yes
Test 8	95	Yes	89	Yes
Test 9	141	Yes	91	Yes
Test 10	125	Yes	110	Yes
Test 11	136	Yes	73	Yes
Test 12	99	No	70	Yes
Test 13	146	Yes	114	Yes
Test 14	189	Yes	94	Yes
Test 15	105	Yes	93	Yes
Test 16	150	Yes	98	Yes
Test 17	156	No	70	Yes
Test 18	178	Yes	90	Yes
Test 19	154	Yes	70	Yes
Test 20	97	No	90	Yes
Test 21	166	No	71	Yes
Test 22	262	No	73	Yes
Test 23	92	Yes	90	Yes
Test 24	125	Yes	126	Yes
Test 25	191	No	91	Yes
Test 26	190	Yes	126	Yes

[**Table of Values obtained from the tests**]

	Token Randomly Generated		Token Circularly Generated	
	Execution Time	Success	Execution Time	Success
Test 27	185	Yes	94	Yes
Test 28	178	Yes	89	Yes
Test 29	129	No	110	Yes
Test 31	148	Yes	110	Yes
Test 32	99	Yes	114	Yes
Test 33	135	No	90	Yes
Test 34	152	Yes	89	Yes
Test 35	130	Yes	110	Yes
Test 36	105	No	114	Yes
Test 37	118	No	90	Yes
Test 38	163	Yes	108	Yes
Test 39	152	Yes	89	Yes
Test 40	191	No	108	Yes
Test 41	178	Yes	126	Yes
Test 42	80	Yes	90	Yes
Test 43	139	Yes	110	Yes
Test 44	190	No	89	Yes
Test 45	105	Yes	108	Yes
Test 46	178	Yes	90	Yes
Test 47	118	Yes	110	Yes
Test 48	215	No	89	Yes
Test 49	170	Yes	79	Yes
Test 50	135	No	90	Yes
Mean	140.78		95.3	
Standard Deviation	39.57		16.77	

3 Final Analysis

3.1 Token Randomly Generated

To assess the performance of our codes, we used a t-test analysis of data from 50 tests for each program in the usual situation. This t-test allowed us to see if there was a significant difference in performance between my script and the script of my colleague. The following parameters must be considered:

Using these to values, the critical value is calculated using a numeric table that returns the critical values associated with a certain degree of freedom and alpha.

- α: We chose an alpha level of 0.05, which translates to a 5 % chance of rejecting the null hypothesis when it is true. This significance level assisted us in determining whether to accept or reject the null hypothesis.

- Df= the number of degrees of freedom, which is determined by the number of tests run by me and my colleague. The formula to compute DF is the following:

$$df = n1 + n2 - 2 = 50 + 50 - 2 = 98 \tag{1}$$

- Critical value: using the estimated t-statistic and degrees of freedom, we derived the critical value from the t- distribution table for a two-tailed test with a significance level (alpha) of 0.05. The critical value is approximately 1.984.

At this point it is possible to compute the t value knowing that my colleague had mean1=161.68 and standard deviation s1 of 78.80. Thus, it is given by:

$$t = \frac{m\bar{e}an_1 - m\bar{e}an_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = 1.88299 \tag{2}$$

3.2 Token Circularly Generated

The same analysis is performed on the token that is generated in a circular manner, and the t value is computed taking into account the values from my work and my colleague- so values now are:

- α: We chose an alpha level of 0.05, which translates to a 5 % chance of rejecting the null hypothesis when it is true. This significance level assisted us in determining whether to accept or reject the null hypothesis.

- Df= the number of degrees of freedom, which is determined by the number of tests run by me and my colleague. The formula to compute DF is the following:

$$df = n1 + n2 - 2 = 50 + 50 - 2 = 98 \tag{3}$$

- Critical value: using the estimated t-statistic and degrees of freedom, we derived the critical value from the t- distribution table for a two-tailed test with a significance level (alpha) of 0.05. The critical value is approximately 1.984.

At this point it is possible to compute the t value knowing that my colleague had mean1=106.94 and standard deviation s1 of 2.94. Thus, it is given by:

$$t = \frac{m\bar{e}an_1 - m\bar{e}an_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = 3.2778 \tag{4}$$

3.3 Successes and Failures

Following the computation of the t-test values, it is crucial to focus on the task's failures and achievements. Using my colleague's data, I created the following table of total successes and failures:

The most prevalent sort of error is the issue of pushing or stripping away an unwanted object, such as a token that has already been set. This mistake causes the silver token to be separated from the golden matched token, but the robot is unconcerned because it is coded that once releasing an object, it does not grasp it again. For example:

random-token	Task Completed	Failures	Error%
my-script	34	16	32%
colleague-script	22	28	56%

circular-token	Task Completed	Failures	Error%
my-script	50	0	0
colleague-script	42	8	16%

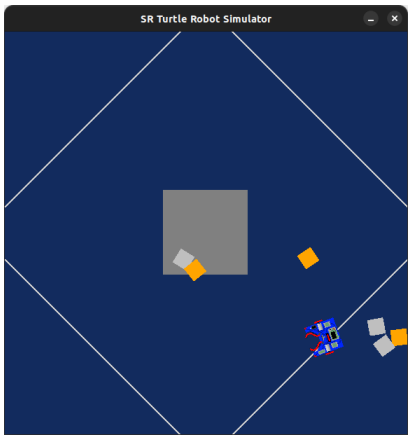

Figure 1: Common error: dragging away an outer silver token causing difficulties on releasing the grabbed one to the golden token

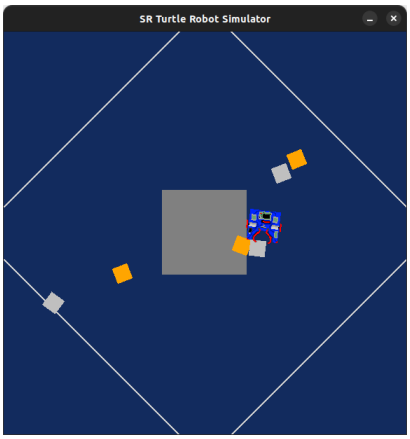

Figure 2: Undesired final state: the bottom-left silver token is far away from the golden paired one but still the output of the script says it is successfully ended

4 Conclusion

In this section are commented the final results:

4.1 Token Randomly Generated - Conclusion

In the random case, the t-value, which is 1.88299, is less than the critical value of 1.984 . That is, as the Null hypothesis is accepted at this point, the difference in performance between my script and the one of my colleague is not statistically significant, and thus it is not possible to make conclusions about the difference in performance of my colleague's script to mine from a time computational standpoint. Nonetheless, the error rate is significantly larger than mine, which has an impact on task performance because it is beyond 50

4.2 Token Circular Generated - Conclusion

In this situation, the result is different since the t-value is 3.2778, which is much higher than the critical t value of 1.984. Though my script has a shorter mean processing time, my colleague obtained a very low number of standard deviation, indicating that his script is more stable for this task, albeit being slightly slower. On the other side, the scenario becomes discernible when examining the error rate on this work, as the algorithm created by my colleague occasionally fails, whilst mine has not failed the task during the tests. Nonetheless, my standard deviation is substantially bigger, indicating that my performance speed is a little unpredictable.

4.3 Final thoughts

In conclusion, considering the two scenarios, my script is more robust in cases when randomness occurs, i.e. when the road to the goal is less linear. The main disadvantage could be an excessive rigidity of the mobile robot's movements while performing its duty, negatively impacting time performance while assuring a minimal risk of failure. On the other hand, my colleague's script works better in a more linear environment, where the token distribution is more uniform, but suffers when randomness arises. A new algorithm that gives better spatial awareness in order to avoid critical and failed errors could be a significant improvement for both of the scripts.